Desafio Titanic

Desafio KAGGLE Titanic

O desafio consiste em a partir dos dados dos passageiros do Titanic, aplicar um modelo de Machine Learning

Para fazer a previsao de sobreviventes, e testar a acuracidade desse modelo.

Bibliotecas que serão usadas

```
library(tidyverse)
library(na.tools)
library(data.table)
library(caTools)
library(e1071) #3 NAIVE BAYES
library(caret)
```

BAIXANDO OS ARQUIVOS DE TESTE E TRINAMNETO TITANIC.

```
base_treinamneto<-fread("train.csv",sep = "auto",sep2 = "auto",integer64 = "numeric",encoding = "UTF-8"
base_teste<-fread("test.csv",sep = "auto",sep2 = "auto",integer64 = "numeric",encoding = "UTF-8" )

base_treinamneto<-as_tibble(base_treinamneto)
base_teste<-as_tibble(base_teste)


base_treinamneto<-base_treinamneto[,c(1,3:12,2)]
```

PRÉ - PROCESSMENTO

IDENTIFICANDO VALORES FALTANTES NAS NASES DE DADOS(NAs)

```
sum(is.na(base_treinamneto))
```

```
## [1] 177
```

```
sum(is.na(base_teste))
```

```
## [1] 87
```

```
quantidade<-NULL
for (i in 1 : length(base_treinamneto)) {
  quantidade[i]<-sum(is.na(base_treinamneto[i]))
}
### 177 NAs na avriavel Age


quantidade2<-NULL
for (i in 1 : length(base_teste)) {
  quantidade2[i]<-sum(is.na(base_teste[i]))
}

quantidade ## BASE_TREINAMNETO 177 NAs
```

```
##  [1]   0   0   0   0 177   0   0   0   0   0   0   0
```

```
quantidade2 ## BASE_TESTE 87 NAs
```

```
##  [1]  0  0  0  0 86  0  0  0  1  0  0
```

Extarir a media de idade por classe e preencher VALORES FALTANTES (NAS)

IREI AJUNTAR AS VARIAVEIS(COLUNAS) DAS DUAS BASES(TESTE E TREINAMENTO)

PARA EXTRAIR A MEDIA DE IDADES POR CLASSES DO NAVIO E FAZER A IMPUTAÇAO DE

DADOS FALTANTES DAS IDADES.

```
base_total<-cbind(c(base_treinamneto$Pclass,base_teste$Pclass))
base_total2<-cbind(c(base_treinamneto$Age,base_teste$Age))
duas_variaveis<-data.frame(base_total,base_total2)
duas_variaveis<-as_tibble(duas_variaveis)
duas_variaveis
```

```
## # A tibble: 1,309 x 2
##    base_total base_total2
##         <int>       <dbl>
##  1          3          22
##  2          1          38
##  3          3          26
##  4          1          35
##  5          3          35
##  6          3          NA
##  7          1          54
##  8          3           2
##  9          3          27
## 10          2          14
## # ... with 1,299 more rows
```

AGORA QUE AS DUAS BASES ESTAO JUNTAS. EXTRAIR A MEDIA DE IDADES POR CLASSES.

```
duas_variaveis %>% group_by(base_total) %>% summarize(media_por_classe=mean(base_total2,na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   base_total media_por_classe
##        <int>            <dbl>
## 1          1             39.2
## 2          2             29.5
## 3          3             24.8
```

USAR "FOR" PARA IMPUTAR AS MEDIAS DE IDADE AOS VALORES FALTANTES DE ACORDO.

COM SUAS RESPECTIVAS CLASSES. PARA BASE_TESTE E BASE_TREINAMENTO.

```
for (i  in 1 : nrow(base_treinamneto)) {
  if (base_treinamneto$Pclass[i] == 3 && is.na(base_treinamneto$Age[i])) {
    base_treinamneto$Age[i] <-24.8

  }
  if (base_treinamneto$Pclass[i] == 2 && is.na(base_treinamneto$Age[i])) {
    base_treinamneto$Age[i] <-29.5
  }
  if (base_treinamneto$Pclass[i] == 1 && is.na(base_treinamneto$Age[i])) {
    base_treinamneto$Age[i] <-39.2
  }
}

for (i  in 1 : nrow(base_teste)) {
  if (base_teste$Pclass[i] == 3 && is.na(base_teste$Age[i])) {
    base_teste$Age[i] <-24.8

  }
  if (base_teste$Pclass[i] == 2 && is.na(base_teste$Age[i])) {
    base_teste$Age[i] <-29.5
  }
  if (base_teste$Pclass[i] == 1 && is.na(base_teste$Age[i])) {
    base_teste$Age[i] <-39.2
  }
}
```

VERIFICANDO QUE NÃO HÁ MAIS VALORES FALTANTES NAS IDADES.

```
base_treinamneto %>% filter(is.na(Age))
```

```
## # A tibble: 0 x 12
## # ... with 12 variables: PassengerId <int>, Pclass <int>, Name <chr>,
## #   Sex <chr>, Age <dbl>, SibSp <int>, Parch <int>, Ticket <chr>, Fare <dbl>,
## #   Cabin <chr>, Embarked <chr>, Survived <int>
```

```
base_teste %>% filter(is.na(Age))
```

```
## # A tibble: 0 x 11
## # ... with 11 variables: PassengerId <int>, Pclass <int>, Name <chr>,
## #   Sex <chr>, Age <dbl>, SibSp <int>, Parch <int>, Ticket <chr>, Fare <dbl>,
## #   Cabin <chr>, Embarked <chr>
```

```
base_teste %>% group_by(Pclass) %>% summarize(media_tarifa=mean(Fare, na.rm = TRUE))
```

VARIAVEL *FARE* DA BASE TESTE POSSUI UM NA. IDENTIFICAR A MEDIA DO VALOR DA PCLASS 3 E FAZER A IMPUTAÇAO.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   Pclass media_tarifa
##    <int>        <dbl>
## 1      1         94.3
## 2      2         22.2
## 3      3         12.5
```

FAZER A IMPUTAÇÃO.

```
base_teste$Fare<-ifelse(is.na(base_teste$Fare),12.5,base_teste$Fare)
### NAS PREENCHIDA COM 12.5
```

TRANSFORMANDO AS VARIAVEIS EM VARIAVEIS CATEGORICAS.

```
base_treinamneto$Pclass<-as_factor(base_treinamneto$Pclass)
base_treinamneto$Sex<-as_factor(base_treinamneto$Sex)
base_treinamneto$Embarked<-as_factor(base_treinamneto$Embarked)
base_treinamneto$Survived<-as_factor(base_treinamneto$Survived)

levels(base_treinamneto$Sex)<-c(1:2)

ID_treinamento<-base_treinamneto$PassengerId
```

```
base_teste$Pclass<-as_factor(base_teste$Pclass)
base_teste$Sex<-as_factor(base_teste$Sex)
base_teste$Embarked<-as_factor(base_teste$Embarked)

levels(base_teste$Sex)<-c(1:2)


ID_teste<-base_teste$PassengerId
```

```
base_treinamneto<-base_treinamneto[,c(2,4:5,9,11:12)]

base_teste<-base_teste[,c(2,4:5,9,11)]
base_teste
```

SELECIONANDO AS VARIAVEIS APARA APLICAR O MACHINE LEARNING.

```
## # A tibble: 418 x 5
##    Pclass Sex     Age  Fare Embarked
##    <fct>  <fct> <dbl> <dbl> <fct>
##  1 3      1      34.5  7.83 Q
##  2 3      2      47    7    S
##  3 2      1      62    9.69 Q
##  4 3      1      27    8.66 S
##  5 3      2      22   12.3  S
##  6 3      1      14    9.22 S
##  7 3      2      30    7.63 Q
##  8 2      1      26   29    S
##  9 3      2      18    7.23 C
## 10 3      1      21   24.2  S
## # ... with 408 more rows
```

```
base_treinamneto
```

```
## # A tibble: 891 x 6
##    Pclass Sex     Age  Fare Embarked Survived
##    <fct>  <fct> <dbl> <dbl> <fct>    <fct>
##  1 3      1      22    7.25 S        0
##  2 1      2      38   71.3  C        1
##  3 3      2      26    7.92 S        1
##  4 1      2      35   53.1  S        1
##  5 3      1      35    8.05 S        0
##  6 3      1      24.8  8.46 Q        0
##  7 1      1      54   51.9  S        0
##  8 3      1       2   21.1  S        0
##  9 3      2      27   11.1  S        1
## 10 2      2      14   30.1  C        1
## # ... with 881 more rows
```

SUMARIZAÇÃO DOS DADOS

```
summary(base_treinamneto)
```

```
##  Pclass  Sex          Age             Fare        Embarked Survived
##  1:216   1:577   Min.   : 0.42   Min.   :  0.00   S:644    0:549
##  2:184   2:314   1st Qu.:22.00   1st Qu.:  7.91   C:168    1:342
##  3:491           Median :26.00   Median : 14.45   Q: 77
##                  Mean   :29.27   Mean   : 32.20    :  2
##                  3rd Qu.:37.00   3rd Qu.: 31.00
##                  Max.   :80.00   Max.   :512.33
```

```
summary(base_teste)
```

```
##  Pclass  Sex          Age             Fare         Embarked
##  1:107   1:266   Min.   : 0.17   Min.   :  0.000   Q: 46
##  2: 93   2:152   1st Qu.:23.00   1st Qu.:  7.896   S:270
##  3:218           Median :25.00   Median : 14.454   C:102
##                  Mean   :29.51   Mean   : 35.572
##                  3rd Qu.:36.38   3rd Qu.: 31.472
##                  Max.   :76.00   Max.   :512.329
```

TREINANDO O ALGORITMO NAIVE BAYES.

```
library(e1071)
classificadorNaive<-naiveBayes(base_treinamneto[,1:5],y = base_treinamneto$Survived)
```

O ALGORITMO NAYVE BAYES CRIA UMA TABELA DE PROBABILIDADES

COM ESSA TABELA ELA GERA O MODELO DE PREVISAO.

```
classificadorNaive
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = base_treinamneto[, 1:5], y = base_treinamneto$Survived)
##
## A-priori probabilities:
## base_treinamneto$Survived
##         0         1
## 0.6161616 0.3838384
##
## Conditional probabilities:
##                           Pclass
```

```
## base_treinamneto$Survived         1         2         3
##                          0 0.1457195 0.1766849 0.6775956
##                          1 0.3976608 0.2543860 0.3479532
##
##                           Sex
## base_treinamneto$Survived         1         2
##                          0 0.8524590 0.1475410
##                          1 0.3187135 0.6812865
##
##                           Age
## base_treinamneto$Survived     [,1]     [,2]
##                          0 29.77923 12.75918
##                          1 28.44933 13.98354
##
##                           Fare
## base_treinamneto$Survived     [,1]     [,2]
##                          0 22.11789 31.38821
##                          1 48.39541 66.59700
##
##                           Embarked
## base_treinamneto$Survived           S           C           Q
##                          0 0.777777778 0.136612022 0.085610200 0.000000000
##                          1 0.634502924 0.271929825 0.087719298 0.005847953
```

APLICANDO O MODELO DE PREVISAO NAIVE BAYES EM NOSSA BASE_TESTE DE DADOS.

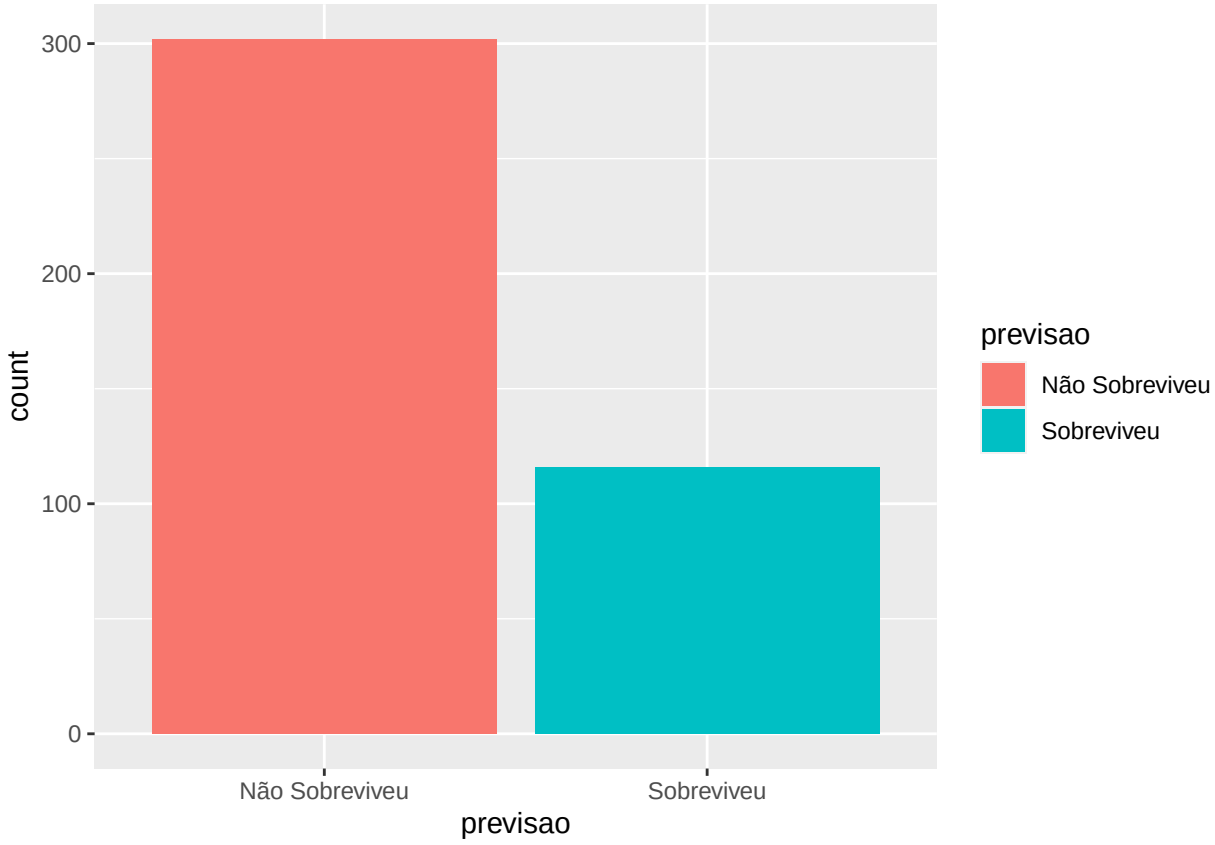
```
previsao<-predict(classificadorNaive, base_teste)
prev<-data.frame(ID_teste,previsao)
as_tibble(prev)
```

```
## # A tibble: 418 x 2
##    ID_teste previsao
##       <int> <fct>
##  1      892 0
##  2      893 0
##  3      894 0
##  4      895 0
##  5      896 0
##  6      897 0
##  7      898 0
##  8      899 0
##  9      900 1
## 10      901 0
## # ... with 408 more rows
```

```
## ONDE 0=NAO SOBREVIVEU 1=SOBREVIVEU
```

GRÁFICO QUE O MODELO DE PREVISÃO GEROU.

```
prev[2]<-as_factor(prev[2])
levels(prev$previsao)<-c("Não Sobreviveu","Sobreviveu")
ggplot(data = prev)+
  geom_bar(mapping = aes(x = previsao, fill=previsao))
```



NO DESAFIO KAGGLE A PREVISAO UTILIZANDO NAYVE BAYES APRESENTOU UMA PONTUAÇAO DE 0.76076